IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPLICANT:	Young-Chun Kim	EXAMINER: Patel, Gautam.
SERIAL NO.:	09/338,473	GROUP ART UNIT: 2655
FILED:	June 22, 1999	Docket: 8836-116
FOR:	APPARATUS FOR CONTROLLING MULTI-WORD STACK OPERATION IN DIGITAL DATA PROCESSORS	

Mail Stop: Appeal Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

Enclosed please find APPELLANTS' APPEAL BRIEF in triplicate.

The amount of $340.00 is hereby authorized to be charged to a credit card. Form PTO-2038 (Credit Card Payment Form) is enclosed. Please charge any deficiency as well as any other fee(s) which may become due under 37 C.F.R. § 1.16 and/or 1.17, at any time during the pendency of this application, or credit any overpayment of such fee(s), to Deposit Account No. 50-0679.

Respectfully submitted,

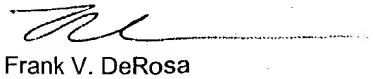
Frank V. DeRosa
Reg. No. 43,584
Attorney for Applicant(s)

F. CHAU & ASSOCIATES, LLC
130 Woodbury Road
Woodbury, NY 11797
Tel: (516) 692-8888
Fax: (516) 692-8889

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

I hereby certify that this correspondence and accompanying documents are being deposited with the United States Postal Service as first class mail, postpaid in an envelope, addressed to the: Mail Stop Appeal Brief Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on October 14, 2004.

Dated: October 14, 2004



Frank V. DeRosa

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicants: Kim et al.

Examiner: Patel, G.

Serial No.: 09/338,473

Group Art Unit: 2655

Filed: June 22, 1999

Docket: 8836-116

For: **APPARATUS FOR CONTROLLING MULTI-WORD STACK OPERATIONS IN DIGITAL DATA PROCESSORS**

APPEAL BRIEF

Appeal from Group 2655

F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, New York 11797
TEL: (516) 692-8888
FAX: (516) 692-8889
Attorneys for Appellant

TABLE OF CONTENTS **Page(s)**

I. INTRODUCTION

This Appeal is from a Final Office Action mailed on December 11, 2003 (Paper No. 14) (hereinafter, referred to as the "FINAL ACTION") finally rejecting claims 1-8 of the above-identified application. Applicants commenced this Appeal by a Notice of Appeal filed on June 14, 2004, and hereby submit this Appeal Brief in furtherance of the Appeal.

II. REAL PARTY IN INTEREST

The real party in interest for the above-identified application is Samsung Electronics Co., LTD., the assignee of the entire right, title and interest in and to the subject application by virtue of an assignment of recorded in the U.S. Patent and Trademark Office at reel/frame 010057/0008.

III. RELATED APPEALS AND INTERFERENCES

There are no Appeals or Interferences known to Applicant, Applicant's representatives or the Assignee, which would directly affect or be indirectly affected by or have a bearing on the Board's decision in the pending Appeal.

IV. STATUS OF CLAIMS

Claims 1-25 are pending. Claims 1-8 stand rejected and are under appeal. Claims 9-25 have been indicated by the Examiner as being directed to allowable subject mater and are not on appeal herein. The claims are set forth in the attached Appendix.

Rejected claims 1, 2 and 6 are independent claims. Claims 3-4 and 7-8 depend directly or indirectly from claims 2 and 6, respectively.

V. STATUS OF AMENDMENTS

An after final amendment was filed on February 17, 2004 in response to the FINAL ACTION, but an Advisory Action mailed on May 20, 2004 (Paper No. 16) indicated that for purposes of Appeal, the Amendment would not be entered.

VI. SUMMARY OF THE INVENTION

The claimed inventions are generally directed to hardware stack memory devices and digital data processors that comprise hardware stack memory devices having multi-bank stack storage frameworks and mechanisms for controlling multi-bank stack storage to enable multi-word PUSH or multi-word POP operations, for example. In particular, in accordance with the claimed inventions, a stack storage is divided into two or more separate memory Banks. A separate Bank Pointer is provided for each memory Bank to point to a location (e.g., top) of the corresponding memory Bank. Control mechanisms are provided for the multi-bank framework to insert bank address data into a plurality of Bank Pointers and generate a control signal to either insert (Push) or remove (Pop) a data word from each of the memory banks at a given time to perform a multi-word Push or multi-word Pop operation.

Independent claims 1, 2 and 6 are representative of claimed inventions that embody the above principles, and are reproduced below for ease of reference:

1. *A hardware stack, comprising:*

a stack storage comprising a plurality of banks each comprising storage locations; and

a stack pointer circuit comprising a bank pointer for each bank, wherein each bank pointer points to a storage location of a corresponding bank, and wherein the stack pointer circuit is responsive to at least one control signal to insert bank address data in at least two bank pointers to perform a multi-word push or multi-word pop operation.

2. *A digital data processor, comprising:*

an instruction decoder for decoding an instruction and generating a plurality of decoding signals;

a stack storage comprising a plurality of banks each comprising storage locations for storing stack items;

a plurality of stack pointers, the stack pointers comprising a main stack pointer for pointing to a top location of the stack storage and a bank pointer for each bank, wherein each bank pointer points to a storage location of a corresponding bank based on the content of the main stack pointer; and

a controller responsive to at least one of the decoding signals for inserting bank address data into at least two bank pointers to perform a multi-word push or multi-word pop operation.

6. *A digital data processor, comprising:*

a stack storage including a plurality of locations, wherein each of the locations of said stack storage is assigned to one of a first bank and a second bank;

a main stack pointer for pointing to a top location of said stack storage;

a first bank stack pointer for pointing to a location assigned to said first bank;

a second bank stack pointer for pointing to a location assigned to said second bank;

an instruction decoder for decoding a stack-based instruction and generating a plurality of decoding signals, each of the plurality of decoding signals denoting one of a one-word push operation, a one-word pop operation, a two-word push operation and a two-word pop operation; and

a stack pointer control logic circuit for controlling said first and second bank stack pointers in response to at least one of the decoding signals to insert bank address data into the first and second bank stack pointers based on the content of the main stack pointer to perform a multi-word push or multi-word pop operation.

VII. ISSUES

Claims 1-8 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,287,309 to Kai in view of U.S. Patent No. 6,167,488 to Koppala.

Thus, the issue on appeal is whether the combination of Kai and Koppala is legally sufficient to establish a *prima facie* case of obviousness against claims 1-8.

VIII. GROUPING OF CLAIMS

For purposes of this Appeal:

(i) Claim 1 stands or falls by itself;

(ii) Claim 2 stands or falls by itself, and dependent claims 3, 4 and 5 stand or fall with claim 2; and

(iii) Claim 6 stands or falls by itself and claims 7 and 8 stand or fall with claim 6.

IX. ARGUMENTS

A. The Combination of Kai and Koppala is *Legally Deficient* to Support a *Prima Facie* Case Of Obviousness Against Claims 1, 2 or 6

In rejecting claims under 35 U.S.C. 103, the Examiner bears the initial burden of presenting a prima facie case of obviousness. In re Rijckaert, 9 F.3d 1531, 1532 (Fed. Cir. 1993). The burden of presenting a prima facie case of obviousness is only satisfied by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references. In re Fine, 837 F.2d 1071, 1074 (Fed. Cir. 1988). A prima facie case of obviousness is established when the teachings of the prior art itself would appear to have suggested the claimed subject matter to

one of ordinary skill in the art. In re Bell, 991 F.2d 781, 782 (Fed. Cir. 1993). The suggestion to combine the references should come from the prior art, and the Examiner cannot use hindsight gleaned from the invention itself to pick and choose among related disclosures in the prior art to arrive at the claimed invention. In re Fine, 837 F.2d at 1075. If the Examiner fails to establish a prima facie case, the rejection is improper and must be overturned. In re Rijckaert, 9 F.3d at 1532 (citing In re Fine, 837 F.2d at 1074).

In the case at bar, Appellants respectfully submit that at the very minimum, the FINAL ACTION does not present any legally sufficient basis for establishing a prima facie case of obviousness of claims 1, 2 or 6 based on the combination of Kai and Koppala. In particular, as explained in detail below, other than bald assertions based on hindsight reasoning, the Examiner has offered no reasonable explanation as to how the combined teachings of Kai and Koppala disclose or suggest the inventions of claims 1, 2 or 6, as a whole. Moreover, as will be demonstrated below, the Examiner has failed to meet his burden of showing why one of ordinary skill in the art would have been motivated to combine the teachings of Kai and Koppala to derive the claimed inventions.

The obviousness rejections for claims 1, 2 and 6 as set forth in the FINAL ACTION are based primarily on the teachings of Kai (particularly, FIG. 2 and relevant description) as modified by the teachings of Koppala. However, as will be explained below, the Examiner's reliance on Kai is wholly misplaced, and even a superficial understanding of the teachings of Kai will readily reveal the impropriety of the obviousness rejections and analysis offered by the Examiner. Therefore, a brief explanation as to the teachings of Kai, as relied on by the Examiner, will first be provided.

In general, Kai discloses (in FIG. 2) a stack memory that is divided into a pair of banks to provide high-speed access operations by simultaneously executing a one word PUSH and a one word POP access operation in advance of determining which access operation should be executed (see, e.g., Col. 1, lines 55-60; and Col. 3, lines 5-12). In particular, Kai discloses in FIG. 2 a stack memory (5) that is divided into two banks, an ODD bank (11) (which is allocated for all odd address) and an EVEN Bank (10) (which is allocated all even addresses). Kai discloses that the data input to both banks is the same (Col. 4, lines 27-43). Kai discloses a method for simultaneously performing a Push (write) and Pop (read) operation with respect to an input address, wherein one bank is used for the Push operation and the other bank is used for the Pop operation. Kai discloses a single stack pointer (6) that stores an address for PUSH data and that the POP data is always stored in an address that is smaller by "1" than the address for storing the PUSH data (see, e.g., Abstract).

Although Kai discloses a stack storage having two banks, the motivation behind using two Bank in the Kai process is that a push (write) and pop (read) operation can be simultaneously performed with respect to an input address, wherein one bank is used for the Push operation and the other bank is used for the Pop operation. The Kai framework enables an effective increase in the stack access time because the stack access operation (read and write) is commenced prior to deciding whether the Push (write) or Pop (read) operation is actually requested with respect to the input address (see, e.g., Abstract, Col 4, lines 10-20). Ultimately, however, only one of the single-word stack accesses (e.g., push or pop) is validated (see, e.g., Abstract, Col. 6, lines 20-32) and the stack pointer SP is adjusted accordingly.

(i) The Combination of Kai and Koppala Does Not Disclose or Fairly Suggest, as a Whole, the Invention of Claims 1, 2 or 6

It is respectfully submitted that the combined teachings of Kai and Koppala do not disclose or fairly suggest, as a whole, the inventions of claims 1, 2 or 6. For instance, the combination does not disclose or suggest hardware stack structures having stack storage that is divided into multiple Banks with separate corresponding Bank Pointers, much less control mechanisms to insert bank address data into multiple Bank Pointers and generate a single control signal to either insert (Push) or remove (Pop) a data word from the separate memory Banks to perform a multi-word Push or a multi-word Pop operation, as essentially claimed in claims 1, 2 and 6.

In general, the obviousness rejections for claims 1, 2 and 6 as set forth in the FINAL ACTION are based primarily on the teachings of Kai (particularly, FIG. 2 and relevant description) as modified by the teachings of Koppala. The Examiner's basis for rejecting claims 1, 2 and 6 is similar, and thus the rejections of such claims will be discussed together. The Examiner's obvious analysis begins on page 2 of the FINAL ACTION by citing FIG. 2 of Kai as disclosing a stack storage comprising a plurality of banks. Although Kai discloses a two-Bank stack storage, the Examiner essentially acknowledges at various sections of the FINAL ACTION that Kai does not disclose performing multi-word push or pop operations using a multi-bank framework. For example, the Examiner acknowledges on Page 3 of the FINAL ACTION that Kai does not disclose that the single stack pointer can be used to perform a multi-word operation. Moreover, the Examiner acknowledges on Page 8 of the FINAL ACTION that Kai does not disclose a PUSH-PUSH or POP-POP operation.

In an effort to cure the deficiencies of Kai in this regard, the Examiner relies on Col. 24, lines 29-32 of Koppala, which generally discloses that "*the most common stack manipulation for stack based computing system is to pop the top two data words off of the stack and to push a data word onto the stop of the stack*". The Examiner further contends (on page 8 of the FINAL ACTION) that Koppala discloses that "*control signal can be used for inserting a two word or multi-word item into said stack storage and removing a two-word item from adjacent locations at a given time*".

The Examiner then points to Kai (Col. 2, lines 18-28), which generally discloses that many variations to the Kai system can be derived by simple mathematical transformation on designation of the addresses. The Examiner then concludes (on page 3 of the FINAL ACTION) that it would have been obvious to one of ordinary skill in the art to have provided, capability to remove two-word item from the stack storage, to the circuit of Kai as taught and suggested by Koppala, because it would have provided a mechanism to execute either a two Push or two POP operation on the stack, thus making the stack operation much faster.

It is respectfully submitted, however, that Koppala does not cure the deficiencies of Kai and that Examiner's grounds for obviousness are nothing more than impermissible hindsight reasoning based on the teachings of the current specification. For instance, although Koppala generally discloses in Col. 24, lines 29-32 removing (popping) two words from the stack, Koppala provides no details (in the cited section) regarding how such multi-word operation is performed (single words separately or simultaneously), much less disclosing or suggesting a multi-word stack operation using a plurality of Banks. Therefore, it is evident that neither Kai nor Koppala expressly discloses using a multi-bank framework to enable multi-word POP or PUSH stack operations, as essentially claimed.

Furthermore, Examiner's reliance on the teaching of Koppala (on page 8 of the FINAL ACTION) that "*control signal can be used for inserting a two word or multi-word item into said stack storage and removing a two-word item from adjacent locations at a given time*", is wholly misplaced. In fact, Koppala's teaching of *removing a two-word item from adjacent locations at a given time* actually teaches against the proposed combination of Kai and Koppala to implement Kai's multi-bank framework with the teachings of Koppala for multi-word push or pop operations.

In particular, as noted above, Kai discloses that the POP data is always stored in an address smaller by "1" than the address for storing the PUSH data (see, e.g., Abstract) In other words, with the two-bank (ODD-EVEN) framework of Kai, the PUSH and POP data are always stored at adjacent locations. This is required in the Kai system in order to implement the simultaneous push and pop operations at the top of the stack, wherein adjacent storage locations (difference of "1") store the data for simultaneous POP and PUSH operations. Therefore, the Examiner's reliance on Koppala as teaching performing a multi-word pop or push "*from adjacent locations at a given time*" is essentially irrelevant with respect to Kai, and actually teaches away from the claimed inventions.

(ii) There Is No Motivation, as a Matter of Law, for Combining the Teachings of Kai and Koppala

In fact, the Examiner basis for motivation to modify the primary reference Kai with the teachings of Koppala to meet the claimed limitations, is legally improper as a matter of law. As noted above, the Examiner essentially contends that it would have been obvious to one of ordinary skill in the art to apply the teachings of Koppala to the circuit of Kai to execute a two

POP operation (removing a two-word item from the stack) instead of one PUSH and one POP operation by simply transforming the logic as suggest by Kai.

However, it is axiomatic that if a proposed modification would render a prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. See, MPEP 2143.01, citing *In re Gordon*, 733 F.2d 900 (Fed. Cir. 1964). Furthermore, if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. See, MPEP 2143.01, citing *In re Ratti*, 270 F.2d 810 (CCPA 1959).

Here, to the extent that Examiner's basis for obviousness is grounded on the stack storage system of Kai being modified by the teachings of Koppala such that Kai can be implemented to execute a two-word PUSH or two-word POP operation using the two-bank stack, instead of a simultaneous single-word PUSH and a single-word POP operation, it is respectfully submitted that such basis is legally improper. Indeed, as noted above, the impetus and basic principle of operation behind the two bank stack system of Kai is to provide a high speed stack operation by simultaneously executing a POP operation in *one bank* and a PUSH operation in the *other bank* so as to access the stack (RAM) before the data process mode is decided (see, e.g., Abstract).

Examiner's proposed modification of the Kai system to perform a two-word PUSH or two-word POP operation - i.e., using each bank to perform a single-word PUSH operation at the same time to obtain a two-word PUSH, or using each bank to perform a single-word POP operation at the same time to obtain a two-word POP - would fundamentally change the principle and purpose of the Kai system. Indeed, in such instance, the Kai system would only be able to perform either a multi-word Read operation or a multi-word Write operation. This

proposed modification of Kai clearly renders Kai unsatisfactory for its intended purpose and significantly changes the principle of operation of the Kai system, which is to *simultaneously* perform both a read operation in one bank and a write operation in the other bank *before* the actual data access mode is determined. As noted above, the two-bank Kai system requires that the POP data is always stored in an address smaller by "1" than the address for storing the PUSH data. Koppala's teaching (as contended by Examiner) of performing a multi-word pop or push " *from adjacent locations at a given time"* in this regard is thus, clearly inconsistent with the teachings of Kai, and not combinable as a matter of law.

Accordingly, claims 1, 2 and 6 are patentable and non-obvious over the combination of Kai and Koppala. Moreover, since claims 3-5 depend from claim 2 and claims 7-8 depend from claim 6, these claims are patentable and non-obvious over the combination of Kai and Koppala at least by virtue of their dependence from respective base claims 2 and 6.

B. CONCLUSION

Accordingly, for at least the above reasons, it is respectfully requested that the Board reverse all claim rejections under 35 U.S.C. § 103(a).

Respectfully submitted,

Frank DeRosa
Reg. No. 43,584

F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, New York 11797
TEL: (516) 692-8888
FAX: (516) 692-8889

APPENDIX A

1. A hardware stack, comprising:

a stack storage comprising a plurality of banks each comprising storage locations; and

a stack pointer circuit comprising a bank pointer for each bank, wherein each bank pointer points to a storage location of a corresponding bank, and wherein the stack pointer circuit is responsive to at least one control signal to insert bank address data in at least two bank pointers to perform a multi-word push or multi-word pop operation.

2. A digital data processor, comprising:

an instruction decoder for decoding an instruction and generating a plurality of decoding signals;

a stack storage comprising a plurality of banks each comprising storage locations for storing stack items;

a plurality of stack pointers, the stack pointers comprising a main stack pointer for pointing to a top location of the stack storage and a bank pointer for each bank, wherein each bank pointer points to a storage location of a corresponding bank based on the content of the main stack pointer; and

a controller responsive to at least one of the decoding signals for inserting bank address data into at least two bank pointers to perform a multi-word push or multi-word pop operation.

3. The digital data processor of claim 2, wherein each location is configured for storing a one-word item.

4. The digital data processor of claim 3, wherein a two-word item is one of inserted into and removed from two adjacent locations at a time.

5. The digital data processor of claim 4, wherein said controller either increases or decreases the content of said main stack pointer by one when the decoding signals indicate a one-word stack operation; and wherein said controller either increases or decreases the content of said main stack pointer by two when the decoding signals indicate a two-word stack operation.

6. A digital data processor, comprising:

a stack storage including a plurality of locations, wherein each of the locations of said stack storage is assigned to one of a first bank and a second bank;

a main stack pointer for pointing to a top location of said stack storage;

a first bank stack pointer for pointing to a location assigned to said first bank;

a second bank stack pointer for pointing to a location assigned to said second bank;

an instruction decoder for decoding a stack-based instruction and generating a plurality of decoding signals, each of the plurality of decoding signals denoting one of a one-word push operation, a one-word pop operation, a two-word push operation and a two-word pop operation; and

a stack pointer control logic circuit for controlling said first and second bank stack pointers in response to at least one of the decoding signals to insert bank address data into the first and second bank stack pointers based on the content of the main stack pointer to perform a multi-word push or multi-word pop operation.

7. The digital data processor of claim 6, wherein said stack storage comprises 2^{n+1} locations, n being a positive integer, and wherein the first bank and the second bank each include 2^n locations.

8. The digital data processor of claim 6, wherein one of the first and second banks includes locations with addresses having a least significant bit of logic >0' and the other of the first and second banks includes locations with addresses having a least significant bit of logic >1'.

9. (Allowed) A digital data processor, comprising:

a stack storage including a plurality of locations, wherein each of the locations of said stack storage are assigned to one of a first bank and a second bank;

a main stack pointer for pointing to a top location of said stack storage;

a first bank stack pointer for pointing to a location assigned to said first bank;

a second bank stack pointer for pointing to a location assigned to said second bank;

an instruction decoder for decoding a stack-based instruction and generating a plurality of decoding signals, each of the plurality of decoding signals denoting one of a one-word push operation, a one-word pop operation, a two-word push operation and a two-word pop operation; and

a stack pointer control logic circuit for controlling said first and second bank stack pointers in response to at least one of the decoding signals to insert bank address data into the first and second bank stack pointers based on the content of the main stack pointer to perform a multi-word push or multi-word pop operation, wherein said stack pointer control logic circuit includes:

an adder for adding one of plurality of predetermined integers to a content of said main stack pointer in response to a first decoding signal from said instruction decoder;

a first selector for selecting for output one of the content of the main stack pointer and a content of said adder in response to a second decoding signal from said instruction decoder, wherein the output of said first selector comprises a high-order bit portion and a low-order bit portion;

a first control logic for generating a first control signal in response to the low-order bit portion of the output from said first selector and a third decoding signal from said instruction decoder;

a second control logic for generating a second control signal in response to the low-order bit portion of the output from said first selector and a fourth decoding signal from said instruction decoder;

an increment logic for incrementing the high-order bit portion of the output from said first selector;

a second selector for selecting one of the high-order bit portion of the output from said first selector and the output of said increment logic in response to the first control signal; and

a third selector for selecting one of the high-order bit portion of the output from said first selector and the output of said increment logic in response to the second control signal;

wherein the outputs of said second and third selectors are provided to said second and first bank stack pointers, respectively.

10. (Allowed) The digital data processor of claim 9, wherein said plurality of predetermined integers include one of +1, +2, -1 and -2; and wherein said increment logic increments the high-order bit portion of the output from said first selector by one.

11. (Allowed) The digital data processor of claim 9, wherein said main stack pointer is updated by the content of said adder.

12. (Allowed) The digital data processor of claim 9, wherein said low-order bit portion of the output from said first selector comprises the least significant bit of the output from the first selector.

13. (Allowed) The digital data processor of claim 12, wherein said first and second control logics alternately enable said second and third selectors, respectively, based on logic states of the least significant bit of the output from the first selector, during the one-word push and pop operations.

14. (Allowed) The digital data processor of claim 13, wherein said second selector is enabled when the least significant bit of the output from said first selector is logic >1'.

15. (Allowed) The digital data processor of claim 13, wherein said third selector is enabled when the least significant bit of the output from said first selector is logic >0'.

16. (Allowed) The digital data processor of claim 12, wherein said first and second control logics enable both of said second and third selectors, irrespective of logic states of the least significant bit of the output from said first selector, during the two-word push and pop operations.

17. (Amended) A digital data processor, comprising:

a stack storage including a plurality of locations, wherein each of the locations of said stack storage are assigned to one of a first bank and a second bank;

a main stack pointer for pointing to a top location of said stack storage;

a first bank stack pointer for pointing to a location assigned to said first bank;

a second bank stack pointer for pointing to a location assigned to said second bank;

an instruction decoder for decoding a stack-based instruction and generating a plurality of decoding signals, each of the plurality of decoding signals denoting one of a one-word push operation, a one-word pop operation, a two-word push operation and a two-word pop operation; and

a stack pointer control logic circuit for controlling said first and second bank stack pointers in response to at least one of the decoding signals to insert bank address data into the first and second bank stack pointers based on the content of the main stack pointer to perform a multi-word push or multi-word pop operation, wherein said stack pointer control logic circuit comprises:

an adder adds one of a plurality of predetermined integers to a high-order bit portion of a content of said main stack pointer in response to a first decoding signal from said instruction decoder;

a control logic for generating one of a first, second, third, and fourth control signals, and combination thereof, in response to a low-order bit portion of the content of said main stack pointer and a second decoding signal from said instruction decoder;

a first selector for selecting one of the high-order bit portion of the content of said main stack pointer and an output of said adder in response to the first control signal;

a second selector for selecting one of the high-order bit portion of the content of said main stack pointer and the output of said adder in response to the second control signal; and

a third selector for selecting one of the high-order bit portion of the content of said main stack pointer and the output of said adder in response to the third control signal;

wherein outputs of said second and third selector are provided to said second and first bank stack pointers, respectively, and the low-order bit portion of the content of said main stack pointer is controlled by the fourth control signal.

18. (Allowed) The digital data processor of claim 17, wherein the plurality of predetermined integers comprise +1 and -1.

19. (Allowed) The digital data processor of claim 17, wherein said high-order bit portion of the content of said main stack pointer is updated by the content of said adder.

20. (Allowed) The digital data processor of claim 17, wherein said low-order bit portion of the content of said main stack pointer comprises a least significant bit of the content of said main stack pointer.

21. (Allowed) The digital data processor of claim 20, wherein said control logic alternately enables said second and third selectors, based on logic states of the least significant bit of the content of said main stack pointer, during the one-word push and pop operations.

22. (Allowed) The digital data processor of claim 21, wherein said second selector is enabled when the least significant bit of the content of said main stack pointer is logic >1'.

23. (Allowed) The digital data processor of claim 21, wherein said third selector is enabled when the least significant bit of the content of said main stack pointer is logic >0'.

24. (Allowed) The digital data processor of claim 17, wherein said control logic enables both said second and third selectors, irrespective of logic states of the least significant bit of the content of said main stack pointer, during the two-word push and pop operations.

25. (Allowed) The digital data processor of claim 20, wherein said control logic toggles the least significant bit of the content of said main stack pointer during the one-word push and pop operations.